**Natural Language Processing**

*Fall 2024*

Project 2: First Deep Learning Project

The purpose of this project is for you to get used to using PyTorch and Google Colab. As such, you are going to use those resources to complete one of the following two tasks:

- Choice 1: Implement an RNN-based text categorization system using Google Colab and PyTorch, and then apply it to one of the three datasets from project #1. This is an example of a sequence classification task.
- Choice 2: Implement an RNN-based POS tagger using Google Colab and PyTorch, and then apply it to a subset of the Penn Treebank. This is an example of a sequence labelling task.

I have posted the Penn Treebank to our class Team, as a gzipped tar file. If you go to the General channel of the Team, then switch to the Files tab, then open the Class Materials folder, you will find it. Please do not share any part of this corpus outside of Cooper Union. The datasets from project #1, of course, have been available from the course website since project #1 was assigned. (I am still not sharing the test sets from the second or third dataset, so if you use one of those, you need to split what I gave you into a training set and a test set, and optionally a tuning set.) Whichever choice you make, you don't need to implement the system from scratch. You can find tutorials online for either type of task and use their starting code. The hardest part of this project will likely be preprocessing the data and creating a proper dataset to be used with your system.

You are not going to be graded on accuracy, as long as it is reasonable. Rather, I want to see that you set things up cleanly and have done some proper experimentation. For example, you might want to compare vanilla RNNs to LSTMs; single-direction LSTMs to bi-LSTMs; stacking two or three layers of LSTMs together; etc. You can also experiment with hyperparameters. Perhaps experiment with learning an embedding layer versus using static embedding such as those produced by word2vec. Do not use transformers for this assignment; I want you to use architectures that we learned about in the second unit of the course.

You should also submit a short writeup that answers the following questions:

- Which of the two tasks did you do?
- Which data did you use, exactly? How did you divide it into a training set and test set (if necessary)? How did you convert the data to the proper format for your system?
- What architectures and hyperparameters did you experiment with? What choices helped or hurt? What setting are used by your final system?
- Does your final architecture learn embeddings for the task, or does it use static word embeddings, and if so, from what method?
- What are the final results? (I should be able to easily run your system and verify this.)
- If you think I need any additional instructions to run your code in Colab, specify this.

When you are finished, share your Colab notebook containing the variation of the architecture that works best with *CarlSable.Cooper@gmail.com* and send me the link. Also send me your

short writeup. The project is due the night of Sunday, November 17, before midnight. (I am not accepting presubmissions for this project; come and talk to me if you have questions.)